INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

PRIORITY DOCUMENT

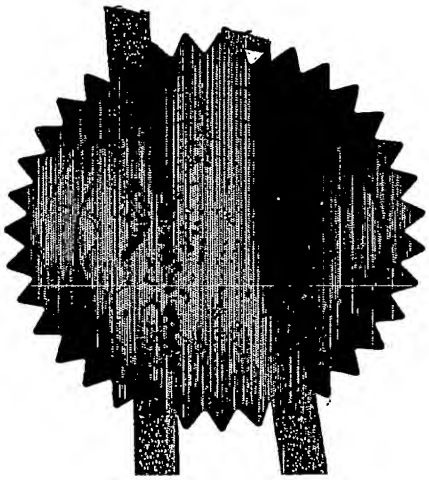
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.



Signed 

Dated 21 September 2004

THE PATENT OFFICE
SW
23 JUN 2003
RECEIVED BY FAX

For Official use only

23JUN03 E817133-1 D10092
0.00-0314593.5

Your reference **Path Remapping (UK)**

0314593.5 23 JUN 2003

The Patent Office

Request for grant of a Patent

Form 1/77 Patents Act 1977

1 Title of invention

A method of enabling an application to access files stored on a storage medium

2 Applicant's details

[X] First or only applicant

2a If applying as a corporate body: Corporate Name

Symbian Limited

Country

GB

2b If applying as an individual or partnership

Surname

Forenames

2c Address

Sentinel House
16 Harcourt Street
London

UK Postcode W1H 1DS

Country GB

ADP Number 7573132002

☐ 2d	Second applicant (if any) Corporate Name Country	
2e	Surname Forenames	
2f	Address UK Postcode Country ADP Number	
3	Address for service	
	Agent's Name	Origin Limited
	Agent's Address	52 Muswell Hill Road London
	Agent's postcode	N10 3JR
	Agent's ADP Number	~~G03274~~ 7270457002

4 Reference Number

Path Remapping (UK)

5 Claiming an earlier application date

An earlier filing date is claimed:

Yes ☐ No ☒

Number of earlier application or patent number

Filing date

15 (4) (Divisional)	8(3)	12(6)	37(4)
☐	☐	☐	☐

6 Declaration of priority

Country of filing	Priority Application Number	Filing Date

7 Inventorship

The applicant(s) are the sole inventors/joint inventors

Yes ☐ No ☒

8 Checklist

			Continuation sheets		
Claims	2	✓	Description	7	✓
Abstract	1	✓	Drawings	0	✓

Priority Documents ~~Yes~~/No

Translations of Priority Documents ~~Yes~~/No

Patents Form 7/77 ~~Yes~~/No

Patents Form 9/77 Yes/~~No~~

Patents Form 10/77 ~~Yes~~/No

9 Request

We request the grant of a patent on the basis of this application

Signed: [signature] Date: 23 June 2003

(Origin Limited)

[redacted]h remapping

A METHOD OF ENABLING AN APPLICATION TO ACCESS FILES STORED ON A STORAGE MEDIUM

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a method of enabling an application, running on an operating system for a portable computing device, to access files stored on a storage medium; the operating system and the storage medium use incompatible directory hierarchies.

2. Description of the Prior Art

Symbian OS applications assume a directory structure that has been defined by Symbian Limited and defines a standard set of directories starting from the root of a drive. Unfortunately this is not compatible with the Memory Stick standard. More specifically, Symbian defines a directory structure on removable drives which contains a number of standard locations, the most basic being:

\

\System

\System\Apps

\System\Libs

\System\Data

\Documents

Only some of these may exist. Some may contain further subdirectories - for example installed applications are placed in a directory below \System\Apps named after the application, and there are various other standard directories.

By contrast, Memory Stick defines a hierarchy like this:

\DCIM

\HIFI

\MSXXX\...

\MSYYY\...

Importantly, the Memory Stick standard says that *only the defined root subdirectories may be placed in the root.* All device-specific data that is not part of the standard must be placed inside one of the MSXXX/MSYYY subdirectories, where the "XXX" "YYY" is a sequence of characters registered with Sony and unique to that device or manufacturer.

Clearly there is a problem, because Symbian defines a directory hierarchy starting at the root but the Memory Stick standard does not allow non-standard directories in the root. To comply with the Memory Stick standard the Symbian hierarchy would have to be inside an MSXXX subdirectory. But all SymbianOS code (including most, if not all, third-party code) has been written assuming the root-based directory structure and cannot easily be modified to use one compliant with Memory Stick.

This leaves a problem of how to create SymbianOS based devices using Memory Stick. We could change all our code so that all paths are completely configurable – or at least a *base directory* can be configured to prefix all application directories. (This base directory would be a MSXXX type registered with Sony.) However this is likely to be a considerable effort both to change the code and also to verify that there aren't any "rogue" cases which create an illegal root directory.

SUMMARY OF THE PRESENT INVENTION

In a first aspect of the invention, a method of enabling an application, running on an operating system for a portable computing device, to access files stored on a storage medium, comprises the following steps:

(a) the application sends a file request with a path that conforms to a directory hierarchy used by the operating system; and

(b) the path in the file request is translated to an equivalent path that conforms to a second directory hierarchy used by the storage medium.

The effect is to map or translate all paths in application file requests to the equivalent path needed by the storage medium. Hence, a path that conforms to the SymbianOS standard can be transparently mapped to a Memory Stick path: it is 'transparent' in that the application has no awareness of the translation that occurs: it simply sees the standard hierarchy mandated by the OS. There is no need to re-write applications to match the requirements of the second storage medium.

A second aspect of the invention is a portable computing device programmed to enable an application running on it to access files stored on a storage medium, in which the application sends a file request with a path that conforms to a directory hierarchy used by the device operating system, the device being further programmed to translate the path in the file request to an equivalent path that conforms to a second directory hierarchy used by the storage medium.

DETAILED DESCRIPTION

The present invention will be described with reference to an implementation for Symbian OS, the operating system for smartphones and other wireless information devices. This implementation enables applications written to run on SymbianOS and using the file hierarchy mandated by SymbianOS to use the Memory Stick storage medium, even though Memory Stick uses an entirely different directory hierarchy.

Root remapping

The requirement is that applications should see a drive (say drive D:) which appears to be a standard Symbian hierarchy but which actually is located somewhere off the root on the Memory Stick. The file system will prefix application file requests with an extra path, called the *root offset*. This happens completely transparently and applications are not aware of the change.

Let's for example say that the root directory registered with Sony is MSSymbian. We want the Symbian hierarchy on drive D: to actually be placed inside the MSSymbian directory. The file system will always add the string "\MSSymbian" to the start of paths.

So for example, take a Memory Stick that has this directory structure:

\
\DCIM
\HiFi
\MSSymbian
\MSSymbian\System
\MSSymbian\Documents

If an application requests a directory listing of "D:*", the file system will internally convert this to "D:\MSSymbian*" and give the result:

\System
\Documents

which is the standard Symbian layout as expected by the application. Note that to the application this *appears* to be the root of the drive but actually it is not.

If the application were now to create a directory "\Documents\MyFiles", this would be translated again by the file system to "\MSSymbian\Documents\MyFiles".

Conceptually the string "\MSSymbian" is prefixed to all strings passed into the file system. In practice this might be implemented in a different way, for example starting all name searches from the MSSymbian directory instead of the root, but the effect is the same.

This therefore allows applications to continue using the Symbian hierarchy but enforces compliance with the Memory Stick standard.

Accessing the root – the magic directory

The root offset method described hides the root completely. Some applications may be Memory Stick aware (that is, they understand the Memory Stick structure and, probably, want to access some of the standard interchange directories defined in the standard). To allow access to the root a "magic" directory is provided, \System\MSROOT. This is really the reverse of the root offset because it is *stripped* from all paths passed to the file system.

So for example if an application wants to access the Memory Stick \DCIM directory (for images), it would use the path "\System\MSROOT\DCIM". The file system would then strip the magic prefix "\System\MSROOT" from this to leave "\DCIM", the intended target directory.

This conversion is only done once so circular references cannot occur. For example the path "\System\MSROOT\MSSymbian\System\MSROOT" appears to be a circular reference, but in fact the file system will only strip the first occurrence of the magic path, to leave "\MSSymbian\System\MSROOT" which doesn't exist and so will return KErrNotFound.

The magic directory doesn't actually exist on the Memory Stick, so if a user was to create a real file or directory \System\MSROOT the magic directory would hide it. It isn't necessary to delete or work around the presence of a real file/directory called \System\MSROOT because the "magic" root mapping is handled entirely within the file system just by modifying the string passed by the client application. The fact that the \System\MSROOT directory/file might exist doesn't prevent the magic root access from working.

However, the user may want to access this file – this is still possible in two ways:

a) Use the "circular reference" \System\MSROOT\MSSymbian\System\MSROOT, which will be converted to \MSSymbian\System\MSROOT on the Memory Stick, which is the user's file/directory.

b) Take advantage of the fact that in SymbianOS the file name is not case-sensitive. If we define that the magic directory *is case sensitive*, then using \System\MSROOT will invoke the magic re-mapping into the root, but any other case, such as \System\msroot, \System\MSRoot, \system\MSROOT will give the user's file/directory.

To avoid applications that search drives from accidentally straying into the magic \System\MSROOT directory and be able to accidentally create files in the root that don't comply with the Memory Stick standard, the magic directory does not appear in a listing of the \System directory content. An application that is Memory Stick-aware would know that it should use \System\MSROOT to access the root. Applications that aren't aware of this will not find it in a directory listing so will not accidentally bypass the enforced SymbianOS directory structure.

Emulating standard directories – ghosting

One further extension is to provide "ghost" directories so that applications that are not specifically Memory Stick aware can still access files from the special Memory Stick directories. Let take as an example a picture viewing application that normally stores its files in

\Documents\Pictures\...

with a number of subdirectories below this which can be named by the user, for example "My Snaps", "Holiday", etc.

The file system can provide another "magic" directory but this time it maps one of the root directories into a directory within the Symbian hierarchy – a *ghost* of the root directory.

So for the example picture viewer, we could create a new ghost directory \Documents\Pictures\MemoryStick that actually maps to \DCIM in the Memory Stick root. The file system in this case is substituting the ghost directory name with the real one.

Thus if the application performs a directory listing of its Pictures directory it will see

My Snaps

Holiday

MemoryStick

And will then show "MemoryStick" as a possible place to find pictures to view. A directory listing of \Documents\Pictures\MemoryStick* will be converted to \DCIM* by the file system and will return the content of the Memory Stick DCIM root directory. The picture viewer can then open any of the files and the same substitution will be done enabling the application to access files from a location it expects while they are actually somewhere else on the Memory Stick.

CLAIMS

1. A method of enabling an application, running on an operating system for a portable computing device, to access files stored on a storage medium, in which the following steps occur:

(a) the application sends a file request with a path that conforms to a directory hierarchy used by the operating system; and

(b) the path in the file request is translated to an equivalent path that conforms to a second directory hierarchy used by the storage medium.

2. The method of Claim 1 in which the storage medium is a storage medium that is removable from the device and conforms to the Memory Stick standard.

3. The method of Claim 2 in which the translation occurs automatically without the application having to be aware of the translation or the existence of a second directory hierarchy.

4. The method of Claim 1 in which the translation is performed by prefixing a file request path to a root of a drive with an extra path to ensure conformance to the second directory hierarchy.

5. The method of Claim 1 in which the translation is performed by stripping away a predefined prefix of a file request path to ensure conformance to the second directory hierarchy.

6. The method of Claim 5 in which stripping away the predefined prefix is only done once per path on the first occurrence of the predefined prefix.

7. The method of Claim 1 in which the translation is performed by mapping a non-existing directory that conforms to the directory hierarchy used by the operating system to a directory that conforms to the second directory hierarchy.

8. The method of Claim 7 in which the mapping allows file interchange to occur.

9. The method of Claim 8 in which the directory that conforms to the second directory hierarchy is a root directory.

10. A portable computing device programmed to enable an application running on it to access files stored on a storage medium, in which the application sends a file request with a path that conforms to a directory hierarchy used by the device operating system, the device being further programmed to translate the path in the file request to an equivalent path that conforms to a second directory hierarchy used by the storage medium.

Abstract

A METHOD OF ENABLING AN APPLICATION TO ACCESS FILES STORED ON A STORAGE MEDIUM

Symbian OS applications assume a directory structure that has been defined by Symbian and defines a standard set of directories starting from the root of a drive. Unfortunately this is not compatible with the Memory Stick standard. The method presented provides applications with a drive that appears to be a standard Symbian drive but actually transparently maps to a safe area on the Memory Stick. It is also possible to access special Memory Stick defined root directories (e.g. for pictures and audio) and to map these directories to standard Symbian-style directories.